

# SM SQZINE

Issue #2

September 2015



Published by:

Timothy Swenson  
 swenson\_t@sbcglobal.net  
 swensont@lanset.com

SMSQzine is published as a service to the Sinclair QL community. Writers are invited to submit articles for publication. Readers are invited to submit article ideas.

Created using Open Source Tools:

- OpenOffice
- Scribus
- Gimp
- SMSQmulator

Copyright 2015  
 Timothy Swenson

Creative Commons License  
 - Attribution  
 - Non-Commercial  
 - Share-Alike

You are free:

- To copy, distribute, display, and perform the work.
- To make derivative works.
- To redistribute the work.

**2nd Annual Midwest Timex/  
 Sinclair Computerfest** 1

**QPC1 & DosBox** 2

**Astronomical Algorithms on the QL** 4

**Computer Graphics with Pascal** 5

**QL Benchmarks** 6

**Editorial** 7

## 2nd Annual Midwest Timex/Sinclair Computerfest

Sinclair entered the American market with the ZX80 and ZX81 through mail order. Timex, who was manufacturing the ZX81, thought it could sell a whole lot of ZX81's through it's dealer network. In July, 1982, the Timex/Sinclair 1000 was introduced.



In November, 1983, the Timex/Sinclair 2068 was introduced as a re-worked Spectrum. With sales less than expected, Timex removed itself from the computer market in March, 1984. When the Sinclair QL came out, those dealers that were still supporting the Timex/Sinclair's stepped up and carried the QL. Unlike the UK, there was not enough QL's sold to create new user groups. The Timex/Sinclair user groups covered all of the Sinclair computers.

So, 3 years after Timex left the market and a year after the QL was discontinued, 700 attendees gathered in a hotel in Indiana to discuss all things



Sinclair, from the ZX81 to the QL. On May 2nd and 3rd, 1987, the Second Annual Midwest Timex/Sinclair ComputerFest was held in

Indianapolis, Indiana. Tickets were \$4 in advanced and \$6 at the door. The event had a dealers room and two smaller rooms set aside for talks. The show ran from 9 am to 6 pm on Saturday and 9 am to 5 pm on Sunday. On Saturday night a banquet was held for the dealers and attendees. Most of the US Sinclair dealers were at the show, along with representatives from 9 user groups.

Some of the key organizers were Paul Holmgren and Frank & Carol Davis, who later formed Mechanical Affinity, a well-known Sinclair dealer.

Mark Steuber of Sharps, Inc, from Virginia was there. Mark was the writer of War in the East, a popular wargame for the QL, Spectrum, and ZX81. Also in attendance was Frank Toemay of Quantum Computer. Frank was known for running ads for the QL in a number of well known trade magazines like Dr. Dobb's Journal and Computer Language.



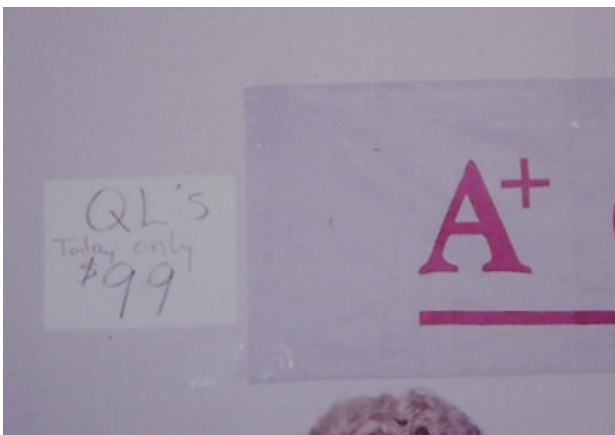
His description of the QL was a little over the top and made it sound like the best thing since sliced bread.

A key QL vendor at the show was A+ Computer Response of New Hampshire with the owners, George and Carol Wilham. Just before Sinclair sold was sold to Amstrad, A+ purchased all US QL's from Sinclair. The Wilhams worked out a deal with Terry Shurwood of Sinclair USA.

The photographs were taken by Tim Woods of Time Designs Magazine and Joe Williamson of Foote Software. They were posted Flickr by the user zxbruno.



One thing of note. A+ Computer Response was selling the base QL for \$179 at the show. One of the pictures shows that they later dropped the price to \$99. Just a year before, the QL was going for \$299 (I know, that is when I bought mine).



## QPC1 and DosBox

Since I've migrated to Linux for my desktop system, my selection of QL emulators is not very big. The best and my favorite is SMSQmulator. It is fairly powerful, but it has limitations in that it is not a QL emulator but a platform for SMSQ/E. This means that very early programs that only run on a QL, will not run on it.

QPC1 has recently been released and that looks to be another option for me. I've heard that some programs that will not run on SMSQmulator, might run better under QPC1. QPC1 is only for DOS, which leaves a bit of a quandary for a Linux user. Luckily, there is DosBox, which creates a DOS

environment, perfect for running QPC1. DosBox is primarily designed for running old DOS games from the early to mid 1990's, but I've used it to run the DOS version of the PSION programs with no issues. DosBox runs on a number of platforms, including Windows, Linux and Mac. It does a good job of providing a DOS environment for running a number of DOS utilities. DosBox is freely available from [www.dosbox.com](http://www.dosbox.com). Download the specific version for your operating system and install it using the directions from the DosBox website.

I will assume that you know where to find QPC1 and be able to download it.

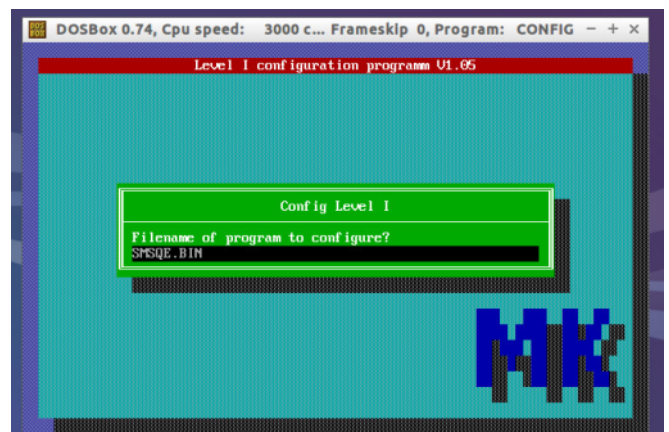
## Configuring DosBox for QPC1

To get QPC1 to run on DosBox, three settings have to change. Open the DosBox config (dosbox.conf) file using a text editor. See the DosBox documentation to find out where your version is located. For Linux, the config file is in the .dosbox directory. Find the EMX, XMS and UMB section of the file and set the following:

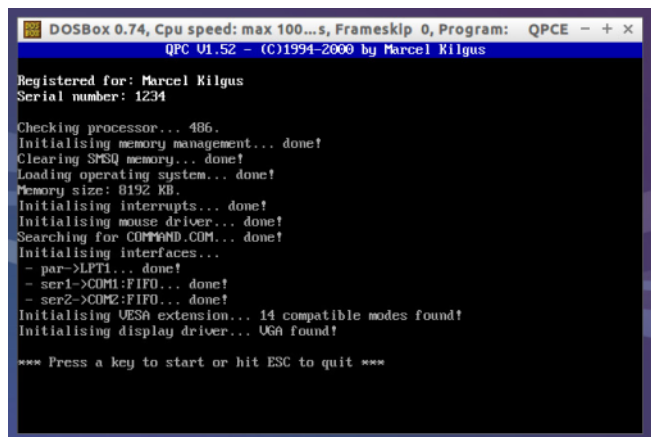
```
xms=false  
ems=false  
umb=false
```

## Setting up DosBox

DosBox can read the underlying file system of the operation system, and can run DOS executables from that file system. It is recommended to create a DOS directory and then place all DOS files in sub directories under this directory. This will keep all of your DOS files in one location. One thing to



remember, once DosBox is started, do not edit the DOS directory with the file system tools. DosBox will not notice any changes until it is restarted again.



When DosBox is started, it will give a Z: drive as the local directory. You can mount the DOS directory from the local file system like this:

```

mount c c:\dos -
Windows
mount c /home/user/dos -
Linux
c: - Move to C:\
  
```

Instead of doing this every time you fire up DosBox, you can add it to the autoexec part of the DosBox config file. Look for [autoexec] in the config file and add the above statements just after it. These commands will be run each time DosBox is started.

## Starting QPC1

Once DosBox is started, just CD to the directory where QPC1 is located. For the first time running QPC1, you will need to run the program "config" and set things like the keyboard and what QXL.WIN files to use. Once that is done, execute 'qpce' for English or 'qpcd' for German.

## Hints with DosBox

### Mouse

By default the mouse is trapped by DosBox, not allowing it to leave the DosBox window. The mouse can be untrapped by hitting CTRL-F10. To

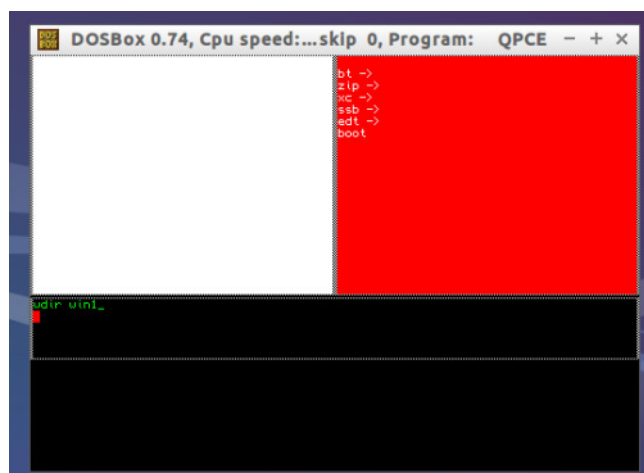
turn this feature off, set "autolock=false" in the DosBox config file.

## Larger Resolution

DosBox allows for higher resolution than VGA. QPC1 also allows for higher resolution. DosBox and QPC1 has been successfully tested with a resolution of 1024x512. See the DosBox and QPC1 documentation on how to change the resolution on each.

## Full Screen

DosBox can be set to full screen by using hitting ALT-ENTER. Hitting this again, will turn off full screen. If using QPC1 in 512x256 mode, going full screen will enlarge all of the text making it easier to see and provide for a more "normal" QL experience.



## Exit from QPC1

For those that don't remember, exiting from QPC1 back to DosBox can be done by the QPC\_EXIT command. DosBox can be exited by clicking on the (X) close box while QPC1 is running, but quitting from QPC1 is probably more graceful. Once back at the DOS prompt, 'exit' will exit DosBox.

## Floppy Access

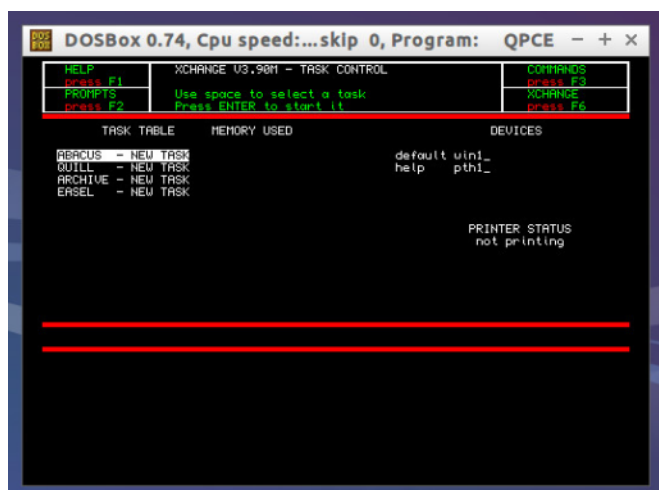
DosBox does not support low level access to floppy disks, even if they are attached, so don't expect to use floppy drives, even though QPC1 can directly access floppies.

## Printer Access

DosBox does not support printers or the printer (LPT) port, which means that even if you used software that captures data to the LPT port and sends it to a file, it will still fail. There is a special build, call Hal9000, of DoxBos that is reported to support the printer port. By printing to a file, it is possible to have to host OS access the file and send it to a printer.

## DOS Access

QPC1 does not have the DOS1\_ device that QPC2 has. There is no way for QPC1 to access files stored under the underlying file system. This also means that there is no built-in way to transfer files



from DOS to a QXL.win file and back. Jonathan Hudson's qxltools is the best option. There is a front-end called wqxt2, but it is only pre-compiled binary for Windows.

Qxltool will format new QXL.WIN files using the command:

```
qxltool -w test.win 2 TEST
```

This will create a 2 Meg QXL.WIN file called test.win with the QDOS label of TEST.

Qxltool also has an interactive mode that will accept a number of commands to manage the QXL.WIN file. You can copy files from the QXL.WIN file to DOS, from DOS to QXL.WIN, create directories, list files, etc. Qxltool is about the only way to get files from DOS to QPC1, without using another emulator.

qxltool for DOS will need to have a copy cwsdpmi.dll. This can be found via a Google search and just needs to be put in the same directory as qxltool.

## Astronomical Algorithms on the QL

In high school, when I was planning for college, I thought I would get a degree in Astronomy. One of the first things I bought with the funds from my first job was a 6-inch Newtonian telescope. I had even join the Astronomy Book of the Month club. Despite getting a degree in Computer Science, I have always been interested in Astronomy.

I've been interested in algorithms that are used for calculating things astronomical. The key book for this is "Astronomical Formulae for Calculators", fourth edition, by Jean Meeus. The first edition was published in 1982 and the fourth in 1988. The follow on book is "Astronomical Algorithms" by Jean Meeus. I have both books and found the first book to be smaller, cheaper and has most of the same details as the second.

The code for each routine is written as closely to the algorithm/calculation in the book. The same variable names are used so that it is easy to read the book and follow exactly what the code is doing. The code examples are in SuperBasic (.bas) and Structured SuperBasic (.ssb).

## Gregorian Date to Julian Date

A Julian Date is the number of days since January 1, 4713 BC. Julian dates are used in Astronomical calculation because it does not have to worry about years, months or days. A Gregorian Date is the day, month, year that we all currently use, from the Gregorian Calendar. The program will take a day, month, and year and convert it to the Julian Date.

## Julian Date to Gregorian Date

This program will take the Julian Date and convert it to a Gregorian Date. Since most of us think in day, month, year, and astronomical calculations are easier done in Julian Date, these two routines are needed for the other programs.

```

Positions of Jupiters' Moons
Input Year (###): 2015
Input Month (##): 7
Input Day (##): 9

Distance from Moons to Jupiter
(in Jupiter radians)
Io      = -5.225617
Europa  = -7.18463
Ganymede = -11.38441
Callisto = -12.89702

      C G E I J

```

## Date of Easter

The date of Easter has been important to the Catholic Church for almost two thousand years. The date of Easter is defined as "the Sunday following the full moon that follows the northern spring equinox."

## Day of Week

Another important calculation is the day of the week when given a Gregorian date. If you want to know what day of the week you were born, this program can calculate that.

```

define function greg2jd(year, month, day)
  local jd, a, b

  if month = 1 or month = 2 then
    year = year - 1
    month = month + 12
  end if

  a = int(year/100)
  b = 2 - a + int(a/4)
  jd =int(365.25*(year+4716))+int(30.6001*
    (month+1))+day+b-1524.5
  return jd
end define

```

## Phase of Moon

The moon is the largest celestial object in the sky. Everyone sees it, but do they really see it. How many know what the current phase it? This program will give one of four phases of the moon for any given date. The calculation is based on the fact that the lunar cycle is 29.53058868 days. With a known full moon date, determining the number of days from that date to the given date, and then dividing by 29.53058868 days, quickly determines the phase.

## Dates of Solstices and Equinox's

A Solstice is the greatest north or south distance that the Sun will progress. It happens in the

Summer and Winter. An Equinox is when the Sun is directly above the equator. This happens in the Spring and Fall. The start of each season is based on either a solstice or equinox. This program will determine the day of each solstice and equinox.

## Positions of Jupiter Moons

The Galilean moons of Jupiter are the 4 that Galileo discovered in 1610, Io, Europa, Ganymede, and Callisto. Using telescope or binoculars, it is fairly easy to make out the 4 moons orbiting around Jupiter. Because they are fairly easy to see with low power optics, they are a common set of objects for beginning stargazers to observe. The moon orbit Jupiter fast enough that the positions change pretty much nightly. This program will calculate the positions of the moons, in relation to Jupiter, on any given date.

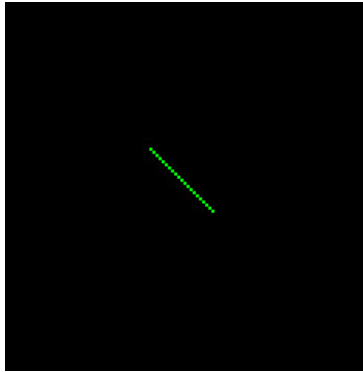
## Computer Graphics with Pascal

When I graduated from college, Pascal was the primary language that was used for most of the classes. Since I've been tinkering with the Computer One Pascal compiler, I started thinking about the books that I kept from college.

One of the last classes that I took on college was Computer Graphics. The primary book we used was "Computer Graphes" by Hearn & Baker. Two of the first algorithms that the book discussed was the line and circle algorithm.

The first primitive for computer graphics is the point (or pixel). There is no algorithm to plot a pixel on the screen, as it is sort of a given. With the point defined, the next algorithm is creating a line, based on drawing a number of pixels.

The most famous algorithm for the line is



Bresenham's. It can draw a line from any two points, to any other two points. It can handle forward and reverse lines, with positive or negative slopes. It is also an algorithm that uses integer math and in the main part of the

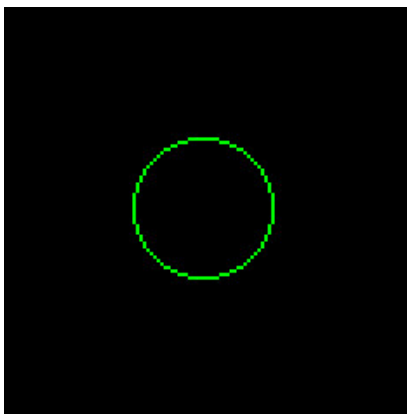
routine, the math calculations are just simple addition. It draws the line from point (x,y) to another point (a,b). If the two end points are switched around, then the drawing goes the opposite way.

I typed in the procedure directly from the text book and the only change I had to make was to substitute the function `set_pixel` with an more QL specific one. With just that one minor change, the procedure compiled just fine with Computer One Pascal. If course, I could have just wrote another routine called `set_pixel` and had it map to the right QL command and that would have made the procedure compile with no changes.

The next primitive is a circle. The algorithm uses a modification of the Bresenham line algorithm to draw the circle in

a slope-like manner. The most obvious method to plot a circle might be to draw a circle using trigonometric functions.

Another might be to use Pythagoras



triangle theorem to draw all of the points. The problem with both of these methods is that they are computationally heavy. In the early days of

computers, they would have taken too long to draw a circle. My using Bresenham's algorithm as a start, the concept that was used for lines was used for circles.

The procedure from the book compiled with no changes, except for the `set_pixel` function, which I had changed as in the line program.

## QL Benchmarks

I have three emulators running on my laptop. Q-Emulator and QPCII are running via Wine, and SMSQmulator is running natively under Java. I was curious to know which one of the emulators were the fastest. Speed is not a critical issue for me, but I was just curious which one of the emulators did have the highest speed. Since all three emulators were running on the same hardware, this takes any CPU speed issues out of the equation. To get the benchmark programs, I went to Dilwyn's side and found some there.

### Bogomips

The first benchmark I ran was Bogomips 1.5, by Thierry Godefroy. Bogomips was designed for the Linux kernel and this version uses the same code as the kernel. With this benchmark the higher the number the better. In this test QPCII came out on top, then Q-Emulator and SMSQmulator.

### Dhrystone

Dhrystone is a benchmark that only uses integer math. The tests that I ran had it go through it's loop for 1 million iterations. The result of the test is a measure of the system in MIPS. QPCII came out with 47.509 MIPS, Q-Emualtor with 37.44 MIPS and SMSQmulator with 22.77 MIPS.

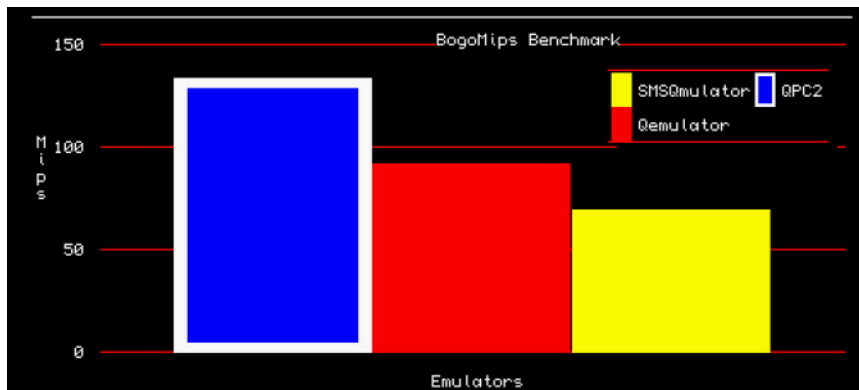
### QSBB

QLAY Superbasic benchmark run a number of routines to test different features of SuperBasic. It covers Print, String, and Functions.

This was the only test in which SMSQmulator beat out Q-Emulator



	<b>QPC2</b>	<b>Q-Emulator</b>	<b>SMSQmulator</b>
Print	174,660	39,080	121,780
Function	712,340	148,580	162,340
Strings	1,010,100	196,840	418,900



was interesting to see the QPCII came out on top of Q-emulator. IIRC, QPCII is written mostly in Assembly, so that would produce faster code.

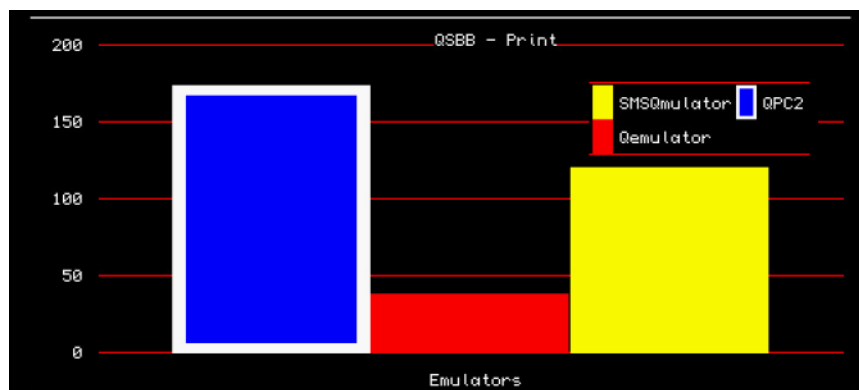
To make the output more interesting, I used Easel to produce the charts. It's been quite a while since I had used Easel and it took a while to get back into how it works.



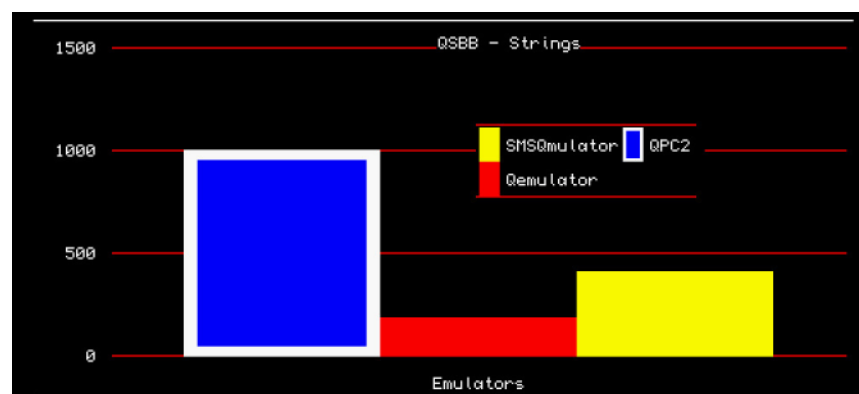
## Editorial

So, why is the editorial at the end of the magazine? Because I started the layout before remembering the editorial.

All of the feedback that I got from the last issue was all positive. I hope the same goes for this issue. The mix of the history, usage and programming articles should provide a little something for everyone. I'm always looking for ideas for articles. Each one of us has a different interest and skill level with the QL. What I might think is common knowledge, might be considered arcane to someone else.



I'll keep this short and get this issue out before September ends.



I knew going into this that SMSQmulator would not be the fastest emulator, given that Java virtual machine is not known for being a speed demon. It